UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/612,193	07/03/2003	Takashi Takenaka	043034-0180	4129

22428 7590 09/25/2006

FOLEY AND LARDNER LLP
SUITE 500
3000 K STREET NW
WASHINGTON, DC 20007

EXAMINER	
ALHIJA, SAIF A	
ART UNIT	**PAPER NUMBER**
2128	

DATE MAILED: 09/25/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/612,193	Applicant(s) TAKENAKA, TAKASHI	
	Examiner Saif A. Alhija	Art Unit 2128	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 July 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *03 July 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *8/20/03*.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-20 have been presented for examination.

PRIORITY

2. Acknowledgment is made of applicant's claim for foreign priority under 35 U.S.C. 119(a)-(d).

Information Disclosure Statement

3. The information disclosure statement (IDS) submitted on 20 August 2003 is in compliance with the provisions of 37 CFR 1.97. Accordingly, the Examiner has considered the IDS as to the merits.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

MPEP 2106 recites:

The claimed invention as a whole must accomplish a practical application. That is, it must produce a "useful, concrete and tangible result" State Street 149 F.3d at 1373, 47 USPQ2d at 1601-02. A process that consists solely of the manipulation of an abstract idea is not concrete or tangibles. See In re Warmerdam, 33 F.3d 1354, 1360, 31 USPQ2d 1754, 1759 (Fed.Cir. 1994). See also Schrader, 22 F.3d at 295, 30 USPQ2d at 1459.

4. **Claims 1-20 are rejected** under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

i) The claims recite a system, method, and computer program. It should be noted that the claims do not appear to produce a useful, concrete, and tangible result since the claims are directed to merely a comparison of cones.

ii) The claims appear to recite a computer program. It should be noted that code (i.e., a computer software program) does not do anything per se. Instead, it is the code stored on a computer that, *when executed*, instructs the computer to perform various functions. The following claim is a generic example of a proper computer program product claim;

A computer program product embodied on a computer-readable medium and comprising code that, when executed, causes a computer to perform the following:

Function A
Function B
Function C, etc...

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. **Claims 1-20 are rejected** under 35 U.S.C. 102(a) as being clearly anticipated by **"Applied Boolean Equivalence Verification and RTL Static Sign-Off", Harry Foster, hereafter referred to as Foster.**

Regarding Claim 1:

Foster discloses A logic verification system comprising: a first logic cone extraction section for extracting first logic cones from a machine-executable object code compiled from a behavioral level description written in a programming language. **(Page 7, Equivalence Proof Techniques and BDD**

Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)

Regarding Claim 2:

Foster discloses The logic verification system according to claim 1, further comprising: a second logic cone extraction section for extracting second logic cones from an RT level description; and a logic cone comparison section for comparing the first logic cones and the second logic cones to verify equivalence between the first and second logic cones. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)**

Regarding Claim 3:

Foster discloses The logic verification system according to claim 1, wherein the first logic cone extraction section includes a symbolic simulation section. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)**

Regarding Claim 4:

Foster discloses The logic verification system according to claim 2, wherein the first logic cone extraction section includes a symbolic simulation section. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1,**

Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)

Regarding Claim 5:

Foster discloses A logic verification system comprising: a storage section for storing an object code compiled from an behavioral level description written in a programming language, an RT level description generated from the behavioral level description, correspondence information which specifies information on pairs of fragments of descriptions to be compared which are included in the behavioral level description and the RT level description and which specifies information on pairs of signals to be compared for each description pair, and compile information including mapping information between the behavioral level description and the object code; a first logic cone extraction section for extracting first logic cones of variables by searching a code portion and the variables of the object code corresponding to each fragments of descriptions and each signals of behavioral level description to be compared which are specified by the correspondence information by referencing the compile information, setting initial symbol values in the variables, performing symbolic simulation from the start to end points of the code portion to produce symbol values when the variable symbolic simulation ends, and using the symbol values as the first logic cones of the variables; a second logic cone extraction section for extracting second logic cones each for the signals for each fragments of description of RT level description to be compared which are specified by the correspondence information; and a logic cone comparison section for comparing the first logic cones and the second logic cones for each signals for each of the fragments of descriptions to be compared in the behavioral level description and the RT level description which are specified by the correspondence information. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and**

Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)

Regarding Claim 6:

Foster discloses A logic verification system comprising: a first logic cone extraction section for extracting first logic cones from a machine-executable object code compiled from an behavioral level description written in a programming language; a storage section for storing properties to be met by the behavioral level description; and a model checking section for checking whether the object code meets the properties. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)**

Regarding Claim 7:

Foster discloses A logic cone extraction apparatus comprising: an input section for inputting an object code compiled from a program description, correspondence information which specifies logic cone extraction areas within the program description and signals to be extracted for each of the logic cone extraction areas, and compile information including mapping information between the program description and the object code; a symbolic simulation section which, by referencing the compile information, searching a code portion and variables of the object code corresponding to logic cone extraction areas and signals to be extracted which are specified by the correspondence information, sets initial symbol values in the variables, and performs symbolic simulation from the start to end points of the code portion; and an output section for outputting symbol values which are obtained when the variable symbolic simulation ends, as logic cones of the variables. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1,**

Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)

Regarding Claim 8:

Foster discloses A logic verification method comprising the step of extracting first logic cones from a machine-executable object code compiled from a behavioral level description written in a programming language. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)**

Regarding Claim 9:

Foster discloses The logic verification method according to claim 8, further comprising the steps of: extracting second logic cones from an RT level description; and comparing the first logic cones and the second logic cones to verify equivalence between the first and second logic cones. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)**

Regarding Claim 10:

Foster discloses The logic verification method according to claim 8, wherein the first logic cones are extracted by performing symbolic simulation. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1,**

Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)

Regarding Claim 11

Foster discloses The logic verification method according to claim 9, wherein the first logic cones are extracted by performing symbolic simulation. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)**

Regarding Claim 12

Foster discloses A logic verification method comprising the steps of: inputting an object code compiled from an behavioral level description written in a programming language, an RT level description generated from the behavioral level description, correspondence information which specifies information on pairs of fragments of descriptions to be compared which are included in the behavioral level description and the RT level description and which specifies information on pairs of signals to be compared for each description pair, and compile information including mapping information between the behavioral level description and the object code; searching a code portion and the variables of the object code corresponding to each fragments of description and each signals of behavioral level description to be compared which are specified by the correspondence information by referencing the compile information; setting initial symbol values in the variables; performing symbolic simulation from the start to end points of the code portion; determining first logic cones of the variables as symbol values when the variable symbolic simulation ends; extracting second logic cones each for the signals for each fragments of RT level description to be compared which are specified by the correspondence information; and comparing

the first logic cones and the second logic cones for each signals for each of the descriptions to be compared in the behavioral level description and the RT level description which are specified by the correspondence information. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)**

Regarding Claim 13

Foster discloses A logic verification method comprising the steps of: extracting first logic cones from a machine-executable object code compiled from an behavioral level description written in a programming language; inputting properties to be met by the behavioral level description; and checking whether the object code meets the properties. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)**

Regarding Claim 14

Foster discloses A logic cone extraction method comprising the steps of: inputting an object code compiled from a program description, correspondence information which specifies logic cone extraction areas within the program description and signals to be extracted for each of the logic cone extraction areas, and compile information including mapping information between the program description and the object code; searching a code portion and variables of the object code corresponding to logic cone extraction areas and signals to be extracted which are specified by the correspondence information by referencing the compile information; setting initial symbol values in the variables; performing symbolic

simulation from the start to end points of the code portion; and outputting symbol values which are obtained when the variable symbolic simulation ends, as logic cones of the variables. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)**

Regarding Claim 15

Foster discloses A computer program instructing a computer to perform logic verification, the program comprising the steps of: a) extracting first logic cones from a machine-executable object code compiled from a behavioral level description written in a programming language; b) extracting second logic cones from an RT level description; and c) comparing the first logic cones and the second logic cones to verify equivalence between the first and second logic cones. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)**

Regarding Claim 16

Foster discloses The computer program according to claim 15, wherein, in the step a), the first logic cones are extracted by performing symbolic simulation. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool**

performance.)

Regarding Claim 17

Foster discloses The computer program according to claim 16, further comprising the step of storing an object code compiled from an behavioral level description written in a programming language, an RT level description generated from the behavioral level description, correspondence information which specifies information on pairs of fragments of descriptions to be compared which are included in the behavioral level description and the RT level description and which specifies information on pairs of signals to be compared for each description pair, and compile information including mapping information between the behavioral level description and the object code, wherein the step a) comprises the steps of: a.1) searching a code portion and the variables of the object code corresponding to each fragments of description and each signals of behavioral level description to be compared which are specified by the correspondence information by referencing the compile information; a.2) setting initial symbol values in the variables; a.3) performing symbolic simulation from the start to end points of the code portion; and a.4) determining the first logic cones of the variables as symbol values when the variable symbolic simulation ends. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)**

Regarding Claim 18

Foster discloses The computer program according to claim 17, wherein the step b) comprises the step of extracting the second logic cones each for the signals for each fragments of RT level description to be compared which are specified by the correspondence information, and the step c) comprises the step of

comparing the first logic cones and the second logic cones for each signals for each of the fragments of descriptions to be compared in the behavioral level description and the RT level description which are specified by the correspondence information. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)**

Regarding Claim 19

Foster discloses A computer program instructing a computer to perform logic verification, the program comprising the steps of: extracting first logic cones from a machine-executable object code compiled from an behavioral level description written in a programming language; inputting properties to be met by the behavioral level description; and checking whether the object code meets the properties based on the first logic cones. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)**

Regarding Claim 20

Foster discloses A computer program instructing a computer to perform logic cone extraction, the program comprising the steps of: inputting an object code compiled from a program description, correspondence information which specifies logic cone extraction areas within the program description and signals to be extracted for each of the logic cone extraction areas, and compile information including mapping information between the program description and the object code; searching a code portion and variables of the object code corresponding to logic cone extraction areas and signals to be extracted which

are specified by the correspondence information by referencing the compile information; setting initial symbol values in the variables; performing symbolic simulation from the start to end points of the code portion; and outputting symbol values which are obtained when the variable symbolic simulation ends, as logic cones of the variables. **(Page 7, Equivalence Proof Techniques and BDD Techniques. Page 9, Column 1, Paragraph 2 and Column 2, Paragraph 2. Page 10, Column 1, Paragraph 1 and Column 2 Paragraph 2. Page 12, Effective equivalence checking methodology and Early Often. Page 13, Combined with RTL recoding to optimize tool performance.)**

6. **Claims 1-20 are rejected** under 35 U.S.C. 102(b) as being clearly anticipated by **"As good as gold", Blackett, hereafter referred to as Blackett.**

Regarding Claim 1:

Blackett discloses A logic verification system comprising: a first logic cone extraction section for extracting first logic cones from a machine-executable object code compiled from a behavioral level description written in a programming language. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 2:

Blackett discloses The logic verification system according to claim 1, further comprising: a second logic cone extraction section for extracting second logic cones from an RT level description; and a logic cone comparison section for comparing the first logic cones and the second logic cones to verify equivalence between the first and second logic cones. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and**

Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)

Regarding Claim 3:

Blackett discloses The logic verification system according to claim 1, wherein the first logic cone extraction section includes a symbolic simulation section. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 4:

Blackett discloses The logic verification system according to claim 2, wherein the first logic cone extraction section includes a symbolic simulation section. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 5:

Blackett discloses A logic verification system comprising: a storage section for storing an object code compiled from an behavioral level description written in a programming language, an RT level description generated from the behavioral level description, correspondence information which specifies information on pairs of fragments of descriptions to be compared which are included in the behavioral level description and the RT level description and which specifies information on pairs of signals to be compared for each description pair, and compile information including mapping information between the behavioral level description and the object code; a first logic cone extraction section for extracting first

logic cones of variables by searching a code portion and the variables of the object code corresponding to each fragments of descriptions and each signals of behavioral level description to be compared which are specified by the correspondence information by referencing the compile information, setting initial symbol values in the variables, performing symbolic simulation from the start to end points of the code portion to produce symbol values when the variable symbolic simulation ends, and using the symbol values as the first logic cones of the variables; a second logic cone extraction section for extracting second logic cones each for the signals for each fragments of description of RT level description to be compared which are specified by the correspondence information; and a logic cone comparison section for comparing the first logic cones and the second logic cones for each signals for each of the fragments of descriptions to be compared in the behavioral level description and the RT level description which are specified by the correspondence information. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 6:

Blackett discloses A logic verification system comprising: a first logic cone extraction section for extracting first logic cones from a machine-executable object code compiled from an behavioral level description written in a programming language; a storage section for storing properties to be met by the behavioral level description; and a model checking section for checking whether the object code meets the properties. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 7:

Blackett discloses A logic cone extraction apparatus comprising: an input section for inputting an object code compiled from a program description, correspondence information which specifies logic cone extraction areas within the program description and signals to be extracted for each of the logic cone extraction areas, and compile information including mapping information between the program description and the object code; a symbolic simulation section which, by referencing the compile information, searching a code portion and variables of the object code corresponding to logic cone extraction areas and signals to be extracted which are specified by the correspondence information, sets initial symbol values in the variables, and performs symbolic simulation from the start to end points of the code portion; and an output section for outputting symbol values which are obtained when the variable symbolic simulation ends, as logic cones of the variables. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 8:

Blackett discloses A logic verification method comprising the step of extracting first logic cones from a machine-executable object code compiled from a behavioral level description written in a programming language. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 9:

Blackett discloses The logic verification method according to claim 8, further comprising the steps of: extracting second logic cones from an RT level description; and comparing the first logic cones and the second logic cones to verify equivalence between the first and second logic cones. **(Page 68,**

Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)

Regarding Claim 10:

Blackett discloses The logic verification method according to claim 8, wherein the first logic cones are extracted by performing symbolic simulation. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 11

Blackett discloses The logic verification method according to claim 9, wherein the first logic cones are extracted by performing symbolic simulation. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 12

Blackett discloses A logic verification method comprising the steps of: inputting an object code compiled from an behavioral level description written in a programming language, an RT level description generated from the behavioral level description, correspondence information which specifies information on pairs of fragments of descriptions to be compared which are included in the behavioral level description and the RT level description and which specifies information on pairs of signals to be compared for each description pair, and compile information including mapping information between the behavioral level description and the object code; searching a code portion and the variables of the object

code corresponding to each fragments of description and each signals of behavioral level description to be compared which are specified by the correspondence information by referencing the compile information; setting initial symbol values in the variables; performing symbolic simulation from the start to end points of the code portion; determining first logic cones of the variables as symbol values when the variable symbolic simulation ends; extracting second logic cones each for the signals for each fragments of RT level description to be compared which are specified by the correspondence information; and comparing the first logic cones and the second logic cones for each signals for each of the descriptions to be compared in the behavioral level description and the RT level description which are specified by the correspondence information. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 13

Blackett discloses A logic verification method comprising the steps of: extracting first logic cones from a machine-executable object code compiled from an behavioral level description written in a programming language; inputting properties to be met by the behavioral level description; and checking whether the object code meets the properties. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 14

Blackett discloses A logic cone extraction method comprising the steps of: inputting an object code compiled from a program description, correspondence information which specifies logic cone extraction areas within the program description and signals to be extracted for each of the logic cone

extraction areas, and compile information including mapping information between the program description and the object code; searching a code portion and variables of the object code corresponding to logic cone extraction areas and signals to be extracted which are specified by the correspondence information by referencing the compile information; setting initial symbol values in the variables; performing symbolic simulation from the start to end points of the code portion; and outputting symbol values which are obtained when the variable symbolic simulation ends, as logic cones of the variables. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 15

Blackett discloses A computer program instructing a computer to perform logic verification, the program comprising the steps of: a) extracting first logic cones from a machine-executable object code compiled from a behavioral level description written in a programming language; b) extracting second logic cones from an RT level description; and c) comparing the first logic cones and the second logic cones to verify equivalence between the first and second logic cones. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 16

Blackett discloses The computer program according to claim 15, wherein, in the step a), the first logic cones are extracted by performing symbolic simulation. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph**

1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)

Regarding Claim 17

Blackett discloses The computer program according to claim 16, further comprising the step of storing an object code compiled from an behavioral level description written in a programming language, an RT level description generated from the behavioral level description, correspondence information which specifies information on pairs of fragments of descriptions to be compared which are included in the behavioral level description and the RT level description and which specifies information on pairs of signals to be compared for each description pair, and compile information including mapping information between the behavioral level description and the object code, wherein the step a) comprises the steps of: a.1) searching a code portion and the variables of the object code corresponding to each fragments of description and each signals of behavioral level description to be compared which are specified by the correspondence information by referencing the compile information; a.2) setting initial symbol values in the variables; a.3) performing symbolic simulation from the start to end points of the code portion; and a.4) determining the first logic cones of the variables as symbol values when the variable symbolic simulation ends. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 18

Blackett discloses The computer program according to claim 17, wherein the step b) comprises the step of extracting the second logic cones each for the signals for each fragments of RT level description to be compared which are specified by the correspondence information, and the step c)

comprises the step of comparing the first logic cones and the second logic cones for each signals for each of the fragments of descriptions to be compared in the behavioral level description and the RT level description which are specified by the correspondence information. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 19

Blackett discloses A computer program instructing a computer to perform logic verification, the program comprising the steps of: extracting first logic cones from a machine-executable object code compiled from an behavioral level description written in a programming language; inputting properties to be met by the behavioral level description; and checking whether the object code meets the properties based on the first logic cones. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

Regarding Claim 20

Blackett discloses A computer program instructing a computer to perform logic cone extraction, the program comprising the steps of: inputting an object code compiled from a program description, correspondence information which specifies logic cone extraction areas within the program description and signals to be extracted for each of the logic cone extraction areas, and compile information including mapping information between the program description and the object code; searching a code portion and variables of the object code corresponding to logic cone extraction areas and signals to be extracted which are specified by the correspondence information by referencing the compile information; setting initial

symbol values in the variables; performing symbolic simulation from the start to end points of the code portion; and outputting symbol values which are obtained when the variable symbolic simulation ends, as logic cones of the variables. **(Page 68, Obstacles in Practice, Paragraph 1. Page 69, Paragraph 1, and A Practical Platform, Paragraph 3-4. Page 70, Paragraph 1, and Equivalence checking in use, Paragraph 1-3. Page 71, Equivalence checking at Cray Research)**

7. **Claims 1-20 are rejected** under 35 U.S.C. 102(b) as being clearly anticipated by **"On the Formal Verification of ATM Switches", Jianping Lu, hereafter referred to as Lu.**

Regarding Claim 1:

Lu discloses A logic verification system comprising: a first logic cone extraction section for extracting first logic cones from a machine-executable object code compiled from a behavioral level description written in a programming language. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 2:

Lu discloses The logic verification system according to claim 1, further comprising: a second logic cone extraction section for extracting second logic cones from an RT level description; and a logic cone comparison section for comparing the first logic cones and the second logic cones to verify equivalence between the first and second logic cones. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 3:

Lu discloses The logic verification system according to claim 1, wherein the first logic cone extraction section includes a symbolic simulation section. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 4:

Lu discloses The logic verification system according to claim 2, wherein the first logic cone extraction section includes a symbolic simulation section. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 5:

Lu discloses A logic verification system comprising: a storage section for storing an object code compiled from an behavioral level description written in a programming language, an RT level description generated from the behavioral level description, correspondence information which specifies information on pairs of fragments of descriptions to be compared which are included in the behavioral level description and the RT level description and which specifies information on pairs of signals to be compared for each description pair, and compile information including mapping information between the behavioral level description and the object code; a first logic cone extraction section for extracting first logic cones of variables by searching a code portion and the variables of the object code corresponding to each fragments of descriptions and each signals of behavioral level description to be compared which are specified by the correspondence information by referencing the compile information, setting initial symbol values in the variables, performing symbolic simulation from the start to end points of the code portion to produce symbol values when the variable symbolic simulation ends, and using the symbol values as the first logic cones of the variables; a second logic cone extraction section for extracting second logic cones each for the signals for each fragments of description of RT level description to be

compared which are specified by the correspondence information; and a logic cone comparison section for comparing the first logic cones and the second logic cones for each signals for each of the fragments of descriptions to be compared in the behavioral level description and the RT level description which are specified by the correspondence information. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 6:

Lu discloses A logic verification system comprising: a first logic cone extraction section for extracting first logic cones from a machine-executable object code compiled from an behavioral level description written in a programming language; a storage section for storing properties to be met by the behavioral level description; and a model checking section for checking whether the object code meets the properties. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 7:

Lu discloses A logic cone extraction apparatus comprising: an input section for inputting an object code compiled from a program description, correspondence information which specifies logic cone extraction areas within the program description and signals to be extracted for each of the logic cone extraction areas, and compile information including mapping information between the program description and the object code; a symbolic simulation section which, by referencing the compile information, searching a code portion and variables of the object code corresponding to logic cone extraction areas and signals to be extracted which are specified by the correspondence information, sets initial symbol values in the variables, and performs symbolic simulation from the start to end points of the code portion; and an output section for outputting symbol values which are obtained when the variable

symbolic simulation ends, as logic cones of the variables. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 8:

Lu discloses A logic verification method comprising the step of extracting first logic cones from a machine-executable object code compiled from a behavioral level description written in a programming language. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 9:

Lu discloses The logic verification method according to claim 8, further comprising the steps of: extracting second logic cones from an RT level description; and comparing the first logic cones and the second logic cones to verify equivalence between the first and second logic cones. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 10:

Lu discloses The logic verification method according to claim 8, wherein the first logic cones are extracted by performing symbolic simulation. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 11

Lu discloses The logic verification method according to claim 9, wherein the first logic cones are extracted by performing symbolic simulation. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 12

Lu discloses A logic verification method comprising the steps of: inputting an object code compiled from an behavioral level description written in a programming language, an RT level description generated from the behavioral level description, correspondence information which specifies information on pairs of fragments of descriptions to be compared which are included in the behavioral level description and the RT level description and which specifies information on pairs of signals to be compared for each description pair, and compile information including mapping information between the behavioral level description and the object code; searching a code portion and the variables of the object code corresponding to each fragments of description and each signals of behavioral level description to be compared which are specified by the correspondence information by referencing the compile information; setting initial symbol values in the variables; performing symbolic simulation from the start to end points of the code portion; determining first logic cones of the variables as symbol values when the variable symbolic simulation ends; extracting second logic cones each for the signals for each fragments of RT level description to be compared which are specified by the correspondence information; and comparing the first logic cones and the second logic cones for each signals for each of the descriptions to be compared in the behavioral level description and the RT level description which are specified by the correspondence information. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 13

Lu discloses A logic verification method comprising the steps of: extracting first logic cones from a machine-executable object code compiled from an behavioral level description written in a programming language; inputting properties to be met by the behavioral level description; and checking

whether the object code meets the properties. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 14

Lu discloses A logic cone extraction method comprising the steps of: inputting an object code compiled from a program description, correspondence information which specifies logic cone extraction areas within the program description and signals to be extracted for each of the logic cone extraction areas, and compile information including mapping information between the program description and the object code; searching a code portion and variables of the object code corresponding to logic cone extraction areas and signals to be extracted which are specified by the correspondence information by referencing the compile information; setting initial symbol values in the variables; performing symbolic simulation from the start to end points of the code portion; and outputting symbol values which are obtained when the variable symbolic simulation ends, as logic cones of the variables. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 15

Lu discloses A computer program instructing a computer to perform logic verification, the program comprising the steps of: a) extracting first logic cones from a machine-executable object code compiled from a behavioral level description written in a programming language; b) extracting second logic cones from an RT level description; and c) comparing the first logic cones and the second logic cones to verify equivalence between the first and second logic cones. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 16

Lu discloses The computer program according to claim 15, wherein, in the step a), the first logic cones are extracted by performing symbolic simulation. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 17

Lu discloses The computer program according to claim 16, further comprising the step of storing an object code compiled from an behavioral level description written in a programming language, an RT level description generated from the behavioral level description, correspondence information which specifies information on pairs of fragments of descriptions to be compared which are included in the behavioral level description and the RT level description and which specifies information on pairs of signals to be compared for each description pair, and compile information including mapping information between the behavioral level description and the object code, wherein the step a) comprises the steps of: a.1) searching a code portion and the variables of the object code corresponding to each fragments of description and each signals of behavioral level description to be compared which are specified by the correspondence information by referencing the compile information; a.2) setting initial symbol values in the variables; a.3) performing symbolic simulation from the start to end points of the code portion; and a.4) determining the first logic cones of the variables as symbol values when the variable symbolic simulation ends. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 18

Lu discloses The computer program according to claim 17, wherein the step b) comprises the step of extracting the second logic cones each for the signals for each fragments of RT level description to be compared which are specified by the correspondence information, and the step c) comprises the step of

comparing the first logic cones and the second logic cones for each signals for each of the fragments of descriptions to be compared in the behavioral level description and the RT level description which are specified by the correspondence information. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 19

Lu discloses A computer program instructing a computer to perform logic verification, the program comprising the steps of: extracting first logic cones from a machine-executable object code compiled from an behavioral level description written in a programming language; inputting properties to be met by the behavioral level description; and checking whether the object code meets the properties based on the first logic cones. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Regarding Claim 20

Lu discloses A computer program instructing a computer to perform logic cone extraction, the program comprising the steps of: inputting an object code compiled from a program description, correspondence information which specifies logic cone extraction areas within the program description and signals to be extracted for each of the logic cone extraction areas, and compile information including mapping information between the program description and the object code; searching a code portion and variables of the object code corresponding to logic cone extraction areas and signals to be extracted which are specified by the correspondence information by referencing the compile information; setting initial symbol values in the variables; performing symbolic simulation from the start to end points of the code portion; and outputting symbol values which are obtained when the variable symbolic simulation ends, as

logic cones of the variables. **(Page 3, Paragraph 3. Page 5, Paragraphs 1-2. Page 8, Paragraph 3. Page 117, Paragraph 1-2.)**

Conclusion

8. The prior art made of record is not relied upon because it is cumulative to the applied rejection. These references include:

i) **U.S. Patent No. 6,611,947.**

9. All Claims are rejected.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Saif A. Alhija whose telephone number is (571) 272-8635. The examiner can normally be reached on M-F, 11:00-7:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kamini Shah can be reached on (571) 272-2279. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

SAA

September 16, 2006

HUGH JONES Ph.D.
PRIMARY PATENT EXAMINER
TECHNOLOGY CENTER 2100